

GetLongPathName

Carefully manage buffer sizes

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-03-23

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6179 bytes

Attack Category	<ul style="list-style-type: none">• Path spoofing or confusion problem	
Vulnerability Category	<ul style="list-style-type: none">• Buffer Management• Indeterminate File/Path	
Software Context	<ul style="list-style-type: none">• File Path Management	
Location	<ul style="list-style-type: none">• winbase.h	
Description	<p>The return buffer for GetLongPathName() and similar functions might return a truncated path and lead to hard-to-find errors.</p> <p>The GetLongPathName() or GetShortPathName() function converts a path to its long or short form. If the lpszLongPath or lpszShortPath buffer is too small to contain the path, the return value is the size of the buffer, in TCHARs, required to hold the path. Call the function again with the proper sized buffer to retrieve the data.</p> <p>The ASCII versions of these functions limit paths to MAX_PATH characters, including the terminating NULL. Unicode versions can handle paths of up to over 32,000 characters if they begin with the special notation "\\?\".</p> <p>GetTempPath() returns the path to the directory for temporary files. It is subject to the same issue regarding potentially returning a path that is too large for the buffer.</p>	
APIs	Function Name	Comments
	GetLongPathName	
	GetLongPathNameA	ASCII implementation
	GetLongPathNameW	Unicode implementation
	GetShortPathName	
	GetShortPathNameA	ASCII implementation
	GetShortPathNameW	Unicode implementation
	GetTempPath	

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	GetTempPathA	ASCII Implementation	
	GetTempPathW	Unicode implementation	
Method of Attack			
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	When any of the indicated functions are called.	It is a best practice to always size path buffers to MAX_PATH characters. For ASCII usage, this guarantees buffers will be large enough. However, for Unicode usage, with some functions this may not always be sufficient. So should also check to confirm that the size was sufficient. For the functions at issue, if the return value is greater than the buffer size, cchBuffer, call the function again with a buffer that is large enough to hold the path.	Effective.
Signature Details	DWORD GetLongPathName(LPCTSTR lpszShortPath, LPTSTR lpszLongPath, DWORD cchBuffer); DWORD GetShortPathName(LPCTSTR lpszLongPath, LPTSTR lpszShortPath, DWORD cchBuffer); DWORD GetTempPath(

	DWORD nBufferLength, LPTSTR lpBuffer);	
Examples of Incorrect Code	<pre> TCHAR shortPath[MAX_PATH] = TEXT("C:\\\\ADIR\\\\SHORT~1.TXT"); LPCTSTR lpszShortPath = shortPath; DWORD buffSize = 15; // Buffer is too small LPTSTR lpszLongPath = (LPTSTR)malloc(buffSize *sizeof(TCHAR)); DWORD result = GetLongPath(shortPath, longPath, buffSize); /* Might have failed - potential bug if don't check result */ </pre>	
Examples of Corrected Code	<pre> TCHAR shortPath[MAX_PATH] = TEXT("C:\\\\ADIR\\\\SHORT~1.TXT"); LPCTSTR lpszShortPath = shortPath; DWORD buffSize = MAX_PATH; LPTSTR lpszLongPath = (LPTSTR)malloc(buffSize *sizeof(TCHAR)); DWORD result = GetLongPath(shortPath, longPath, buffSize); if (result > MAX_PATH) { delete lpszLongPath; buffSize = result; lpszLongPath = (LPTSTR)malloc(buffSize*sizeof(TCHAR)); if (! GetLongPath(shortPath, longPath, buffSize)) { handleError(); } } </pre>	
Source Reference	<ul style="list-style-type: none"> • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/fileio/fs/getlongpathname.asp² 	
Recommended Resources	<ul style="list-style-type: none"> • MSDN reference for GetLongPathName³ • MSDN reference for GetShortPathName⁴ • MSDN reference for GetTempPath⁵ 	
Discriminant Set	Operating System	<ul style="list-style-type: none"> • Windows
	Languages	<ul style="list-style-type: none"> • C • C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>